

## CODE SEQUENCE GENERATOR

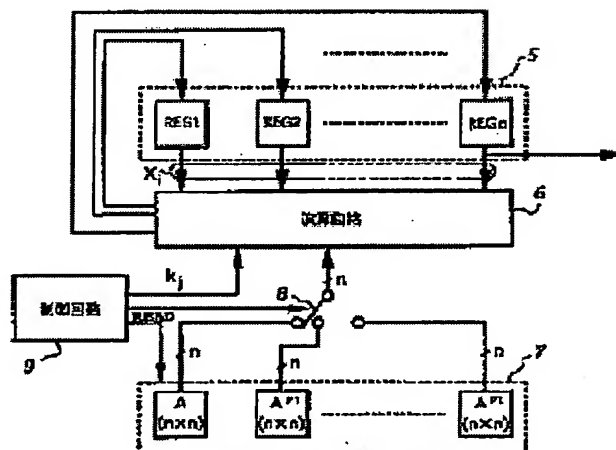
**Patent number:** JP8018550  
**Publication date:** 1996-01-19  
**Inventor:** AZUMA AKIHIRO; ONO HIROSHI; UMEDA SHIGEMI  
**Applicant:** NIPPON TELEGRAPH & TELEPHONE  
**Classification:**  
 - international: H04L9/00; H04L9/10; H04L9/12; G06F7/58  
 - european: H04L9/22  
**Application number:** JP19950087191 19950412  
**Priority number(s):** JP19950087191 19950412; JP19940090353 19940427

Also published as:

EP0680172 (A)  
 US5596516 (A)  
 EP0680172 (A)  
 EP0680172 (B)

## Abstract of JP8018550

**PURPOSE:** To generate a code sequence from an optional code sequence phase by repetitively using an arithmetic procedure of an arithmetic circuit applying arithmetic operation to a register latching a code state while revising the procedure to set a state of the register at a high speed. **CONSTITUTION:** An arithmetic circuit 6 multiplies a received matrix  $AP<j>$  with a state vector  $X_i$  of a register 5. When the matrix  $AP<j>$  is applied to the circuit 6, a control circuit 9 applies a read signal to a memory circuit 7, and then the matrix  $AP<j>$  is fed to the circuit 6 via a selector switch sequentially one by one row each starting from a 1st row. Furthermore, the control circuit 9 allows the circuit 6 to receive an arithmetic coefficient  $K_j$  and obtains a state of the register 5 after the state of  $K_j$  as equation. The matrix  $AP<j>$  is calculated in advance and stored in a memory circuit 7 and the matrix  $AP<j>$  is obtained by reading the content of the circuit 7 and then a value  $(AP<j>)K<j>$  is obtained by multiplying the read matrix  $AP<j>$  by number of  $K_j$  times repetitively. The arithmetic operation of the equation is obtained by multiplication of  $m$ -times ( $m=k_0+K_1+...K_t$ ) and the result is stored in the register 5, from which an output is provided.



$$A \cdot X_i = (A)^{k_0} (AP_1)^{k_1} (AP_2)^{k_2} (AP_3)^{k_3} \dots (AP_t)^{k_t} X_i$$

Data supplied from the esp@cenet database - Worldwide

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平8-18550

(43)公開日 平成8年(1996)1月19日

(51)Int.Cl.<sup>6</sup>

識別記号

庁内整理番号

F I

技術表示箇所

H 0 4 L 9/00

9/10

9/12

G 0 6 F 7/58

C

H 0 4 L 9/ 00

Z

審査請求 未請求 請求項の数5 O L (全 7 頁)

(21)出願番号 特願平7-87191

(22)出願日 平成7年(1995)4月12日

(31)優先権主張番号 特願平6-90353

(32)優先日 平6(1994)4月27日

(33)優先権主張国 日本 (J P)

(71)出願人 392026693

エヌ・ティ・ティ移動通信網株式会社  
東京都港区虎ノ門二丁目10番1号

(72)発明者 東 明洋

東京都港区虎ノ門二丁目10番1号 エヌ・  
ティ・ティ移動通信網株式会社内

(72)発明者 大野 公士

東京都港区虎ノ門二丁目10番1号 エヌ・  
ティ・ティ移動通信網株式会社内

(72)発明者 梅田 成規

東京都港区虎ノ門二丁目10番1号 エヌ・  
ティ・ティ移動通信網株式会社内

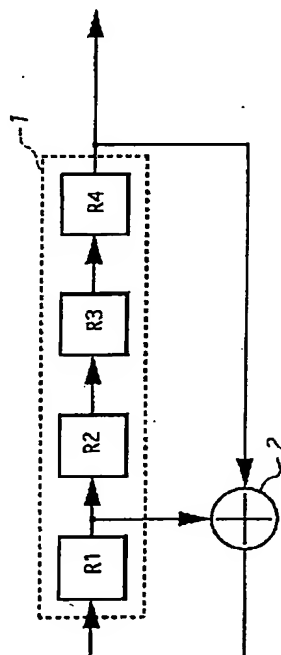
(74)代理人 弁理士 谷 義一 (外1名)

(54)【発明の名称】 符号系列発生器

(57)【要約】

【目的】 周期が非常に長い符号系列であっても、短時間で所望のレジスタ状態を実現し、かつ発生符号のクロックと同じ速度のクロックにより、符号系列を発生できる符号系列発生器。

【構成】 nビットの符号を保持するレジスタ(5)と、該レジスタ(5)に対して行列演算を施す演算回路(6)とを具え、その演算結果を前記レジスタ(5)に帰還することにより逐次的に符号系列を発生する符号系列発生器。演算回路(6)の行列を、メモリ回路(7)とスイッチ(8)とにより変更して繰り返し用いて所望の符号系列を発生する。



## 【特許請求の範囲】

【請求項1】 レジスタに格納された $n$ ビット長のデータ $X$ に、 $n \times n$ 行列 $A$ で規定された線形変換を繰り返して実行し、符号系列 $Y = A^m X$  ( $m$ は正の整数)を発生\*

\* する符号系列発生器において、

予め定めた $t+1$ 個の正の整数 $p_j$  ( $j=0, 1, 2, \dots, t$ )のそれぞれについて、

【外1】

前記行列 $A$ の $p_j$ 乗、すなわち、 $A^{p_j}$ を記憶する記憶手段と、

前記整数 $m$ を、前記整数 $p_j$ の1次式、 $m = k_0 + k_1$

※前記 $k_0, k_1, k_2, \dots, k_t$ の内ではゼロでないものについて、

$p_1 + k_2 p_2 + \dots + k_t p_t$  ( $k_0, k_1, k_2, \dots,$

$k_t$ は0または正の整数)の形に分解する手段と、

※ 【外2】

前記記憶手段から対応する行列 $A^{p_0}, A^{p_1}, A^{p_2}, \dots, A^{p_t}$ を順次選択する

選択手段と、

前記 $k_0, k_1, k_2, \dots, k_t$ の内ではゼロでないもの ★【外3】

について、

★

行列演算 $Y = (A^{p_0})^{k_0} (A^{p_1})^{k_1} (A^{p_2})^{k_2} \dots (A^{p_t})^{k_t} X$ を実行する演算手

段と

を具備することを特徴とする符号系列発生器。

☆【請求項3】 前記選択手段は、並列動作する $n$ 個の $t+1/1$ マルチプレクサを備えたことを特徴とする請求項2に記載の符号系列発生器。

【請求項2】 前記整数 $p_j$ は、 $p_0 = 1, p_1 = n/2, p_2 = n/2^2, p_3 = n/2^3, \dots, p_t = n/2^t$  (ただし、小数点以下は切り捨てて $p_j$ が正の整数となるようにする)であることを特徴とする請求項1に記載の符号系列発生器。

20 【請求項4】 前記選択手段は、

【外4】

☆

前記行列 $A^{p_j}$ の内の選択すべき行列のアドレスを、択一的に前記記憶手段に

供給する

ことを特徴とする請求項2に記載の符号系列発生器。

【請求項5】 前記選択手段は、特定の1行列を選択して前記演算手段に供給することによって、任意の符号間隔の符号行列を発生させることを特徴とする請求項1に記載の符号系列発生器。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、最長系列、Gold系列などの符号系列発生器に関する。

【0002】

【従来の技術】ディジタル符号系列は、ディジタル情報の秘匿、スペクトラム拡散通信における信号帯域拡大のための拡散符号など、広い分野で使用されている。特に最長系列(M系列)は、その周期が非常に長く、また自己相関特性がインパルス的なため、ランダム符号、雑音源として使用される。

【0003】ディジタル符号は、レジスタと、そのレジスタ内容に演算を施す演算回路とを用い、その演算内容をレジスタに帰還させることにより発生させることができる。

【0004】例えば、周期15のM系列の場合、演算が排他的論理和1つで実現でき、帰還法も単純なため、図1に示すような単純な構成とすることができる。

【0005】図1において、1は4つのレジスタR1～R4から成るシフトレジスタ回路、2はレジスタR1と

R4の排他的論理和をとる排他的論理和回路であり、その出力をレジスタR1に帰還する。レジスタR1～R4の各々に初期値を設定し(すべてのレジスタの内容が0の場合を除く)、これらレジスタR1～R4にクロックを与えて、その内容を当該クロックでシフトさせることにより、周期15のM系列符号が発生する。

【0006】図2は、図1の構成を一般化したもので、シフトレジスタ1の代りに各1ビットの4つのレジスタを用いた周期15のM系列発生器の一例を示す。図2において、3は、各1ビットの4つのレジスタREG1～REG4からなるレジスタ回路、4はレジスタ3の内容を受けて演算を施す演算回路であり、その演算結果をレジスタ回路3に帰還させる。M系列符号の出力はレジスタREG4の出力端から得られる。

【0007】これらのM系列符号発生器において、あるレジスタ状態にある時に、任意個数のクロックを与えて、そのレジスタ状態を変化させ、その後から符号を発生させる場合を考える。たとえば、5クロック後から符号を発生させるためには、5クロック分の高速なクロックを与えることにより、スタート時のレジスタ状態にすることができる。この場合に、周期15程度のM系列を、任意の位相から発生させることは容易であるが、周期が非常に長い系列の任意の状態から符号を発生させる場合、該当する個数のクロックにより、希望するレジスタ状態を実現するためには、非常に長い時間が必要とな

る場合がある。

【0008】さらに、生成される符号系列から、1符号おきの符号系列を発生させる場合、従来の技術によれば、2倍のクロックにて符号系列発生器を動作させ、そこで得られる符号のうちの1つおきの符号を選択する必要がある。したがって、 $m$ 符号ごとの符号系列を発生させる場合には、 $m$ 倍のより高速のクロックにて符号系列発生器を動作させる必要がある。このように、従来例では、発生させる符号のクロックよりも高速なクロックが必要である。

【0009】

【発明が解決しようとする課題】このように、従来の技術では、周期が非常に長い符号系列を扱う場合、任意の状態から符号を発生させるためには、非常に長い時間が必要となる。さらにまた、任意の符号ごとの符号系列を発生させる場合には、高速なクロックが必要となる。 \*

前記行列  $A$  の  $p_i$  乗、すなわち、 $A^{p_i}$  を記憶する記憶手段と、

【0013】前記整数  $m$  を、前記整数  $p_i$  の1次式、 $m = k_0 + k_1 p_1 + k_2 p_2 + \dots + k_t p_t$  ( $k_0, k_1, k_2, \dots, k_t$  は0または正の整数) の形に分解する手段と、前記  $k_0, k_1, k_2, \dots, k_t$  の内でゼロ※

前記記憶手段から対応する行列  $A^{p_0}, A^{p_1}, A^{p_2}, \dots, A^{p_t}$  を順次選択する選択手段と、

【0015】前記  $k_0, k_1, k_2, \dots, k_t$  の内でゼロでないものについて、

行列演算  $Y = (A^{p_0})^{k_0} (A^{p_1})^{k_1} (A^{p_2})^{k_2} \dots (A^{p_t})^{k_t} X$  を実行する演算手段と

【0017】を具備することを特徴とする。

【0018】請求項2に記載の発明は、請求項1に記載の符号系列発生器において、前記整数  $p_i$  は、 $p_0 = 1, p_1 = n/2, p_2 = n/2^2, p_3 = n/2^3, \dots, p_t = n/2^t$  (ただし、小数点以下は切り捨てて  $p_i$  が正の整数となるようにする) であることを特徴とする。

【0019】請求項3に記載の発明は、請求項2に記載☆

前記行列  $A^{p_i}$  の内の選択すべき行列のアドレスを、択一的に前記記憶手段に供給することを特徴とする。

【0022】請求項5に記載の発明は、請求項1に記載の符号系列発生器において、前記選択手段は、特定の1行列を選択して前記演算手段に供給することによって、任意の符号間隔の符号行列を発生させることを特徴とする。

【0023】

【作用】本発明によれば、符号系列発生器のレジスタの状態ベクトルを  $X_i$  ( $i$  は正の整数) とすると、1状態後のレジスタ状態は

【0024】

\*【0010】本発明はこのような場合でも短時間で任意所望のレジスタ状態を実現し、しかも、発生させる符号のクロックと同じ速度のクロックにより、任意の符号間隔で符号系列を発生させることのできる符号系列発生器を提供することを目的とする。

【0011】

【課題を解決するための手段】このような課題を解決するために請求項1記載の発明は、レジスタに格納された  $n$  ビット長のデータ  $X$  に、 $n \times n$  行列  $A$  で規定された線形変換を繰り返して実行し、符号系列  $Y = A^m X$  ( $m$  は正の整数) を発生する符号系列発生器において、予め定めた  $t+1$  個の正の整数  $p_i$  ( $i = 0, 1, 2, \dots, t$ ) のそれぞれについて、

【0012】

【外5】

※でないものについて、

【0014】

20 【外6】

★【0016】

★ 【外7】

30 ☆の符号系列発生器において、前記選択手段は、並列動作する  $n$  個の  $t+1/1$  マルチプレクサを備えたことを特徴とする。

【0020】請求項4に記載の発明は、請求項2に記載の符号系列発生器において、前記選択手段は、

【0021】

【外8】

【数1】  $X_{i+1} = AX_i$

で表わされる。ここで、行列  $A$  は演算手順を行列表示したものである。従来の技術によれば、 $m$  状態後のレジスタの状態を求めるには  $X_{i+m} = A^m X_i$ 、つまり  $A$  の演算を  $m$  回行う必要がある。

【0025】本発明における符号系列発生器では、演算回路における掛算の回数が減少可能である。例えば、 $m = k_1 p_1 + k_2 p_2 + k_3 p_3$  で表わされるとする ( $p$  および  $k$  は0以上の整数)。

50 【0026】

【外9】

A,  $A^{p_1}$  および  $A^{p_2}$  という行列をあらかじめ計算しておく、m 状態後のレジスタの状態  $X_{1..m}$  は

【0027】

【数2】

$$X_{1..m} = A^m X_1$$

$$= (A^{p_1})^{k_1} (A^{p_2})^{k_2} (A)^{k_0} X_1$$

【0028】で求めることができる。

【0029】ここで、掛算回数は  $k_1 + k_2 + k_0$  となり、掛算回数を大幅に減少させることができる。たとえば  $m=127$  の場合、 $p_1=100$ ,  $p_2=10$  とすると、 $k_1=1$ ,  $k_2=2$ ,  $k_0=7$  となり演算回数は10となる。mの値が大きくなるほど、演算回数が減少する効果は大きくなる。

【0030】本発明では、このように、複数の行列を組み合わせて使用することにより、高速にレジスタ状態を設定し、任意の符号系列位相から符号系列を発生させることができる。本発明は、逐次的に次の状態を求めるのではなく、離散的に離れたレジスタ状態への演算手順を表わす行列をあらかじめ求めておく点が従来の技術と異なる。

t+1個の  $n \times n$  行列  $A^{p_1}$ ,  $A^{p_2}$ , ...,  $A^{p_t}$  を格納している。

【0037】ここで、値tは、メモリ回路7の容量と、演算の簡単化の程度とを考慮して定められる。たとえば、 $t=4$ ,  $n=100$  とした場合、 $p_j$  ( $j=0, 1, 2, \dots, t (=4)$ ) は、次のように定められる。まず、行列Aは不可欠であるから、 $p_0=1$  とする。ついで、値nを順次2で割って、 $p_1=n/2=50$ ,  $p_2=n/2^2=25$ ,  $p_3=n/2^3=12$ ,  $p_4=p_3/2=6$  と決定される。ただし、小数点以下は※

制御回路9の制御の下に、所望の行列  $A^{p_j}$  を選択し、選択した行列の各行を、順次、その出力端子から演算回路6に供給する。演算回路6は、供給された行列  $A^{p_j}$  と状態ベクトル  $X_1$  とを乗算する回路である。

【0040】さて、状態ベクトル  $X_1$  の1状態後のレジスタ状態  $X_{1..1}$  は、

$$X_{1..1} = A X_1$$

で与えられ、m状態後のレジスタ状態  $X_{1..m}$  は、

【0042】

$$X_{1..m} = A^m X_1$$

で与えられる。

【0043】制御回路9は、このmを  $p_1$  の一次式に分解する。すなわち、

$$m = k_0 + k_1 p_1 + k_2 p_2 + \dots + k_t p_t \quad (3)$$

という形に、mを分解する。たとえば、上述した  $n=100$ ,  $t=4$  の場合、 $m=81$  は、

$$m = 1 \times 50 + 1 \times 25 + 6 \quad (4)$$

\*【0031】さらにまた、本発明によれば、クロック速度を変えずに任意の符号間隔の符号系列を発生させることができる。

【0032】従って、本発明は、符号系列長が長く、大きく離れたレジスタ状態の設定の場合に特に大きな効果がある。

【0033】

【実施例】以下、図面を参照して本発明の実施例を詳細に説明する。

【0034】図3は、本発明による符号系列発生器の一実施例を示すブロック図である。図3において、参照符号5は、nビットのレジスタであり、n個の1ビットレジスタ  $REG1, REG2, \dots, REGn$  から構成されている。レジスタ5は、nビットの状態ベクトル  $X_i$  ( $i$  は0または正の整数) を保持するもので、この  $X_i$  が演算回路6に供給される。

【0035】一方、参照符号7は、メモリ回路であり、

【0036】

\*【外10】

※切り捨てて、 $p_j$  が正の整数となるようにしている。

【0038】メモリ回路7のデータ出力は、nビット単位で、選択スイッチ8のt+1個の入力接点に、それぞれ供給される。選択スイッチ8は、例えば、並列動作するn個のt+1/1マルチプレクサからなり、

【0039】

\*【外11】

★【0041】

★【数3】

(1)

☆【数4】

☆

(2)

◆【0044】

【数5】

◆

\*【0045】

\*【数6】

と分解される。この分解は、 $m$ を2で順次割ることによって、容易に行われる。この分解に基づき、制御回路9は、

\* [0046]  
[外12]

演算に使用される行列 $A^{p_i}$ がメモリ回路7から演算回路6に供給されるように、選択スイッチ8を制御する。そして、制御回路9からメモリ回路7に読み出し信号を供給すれば、行列 $A^{p_i}$ が、第1行から始めて1行ずつ順番に、

【0047】選択スイッチ8を介して、演算回路6に供給される。制御回路9は、また、演算回路6に対して、演算回数 $k_i$ を指示する。

\* [0048]このような構成によれば、(3)式より、  
10 [0049]  
※ [数7]

$$A^m X_i = (A)^{k_0} (A^{p_1})^{k_1} (A^{p_2})^{k_2} (A^{p_3})^{k_3} \dots (A^{p_n})^{k_n} X_i \quad (5)$$

【0050】と表すことができる。この場合、  
【0051】

★ [外13]

$A^{p_i}$ は、予め計算されて、メモリ回路7に格納されているため、これを読み出すだけで得られる。従って、 $(A^{p_i})^{k_i}$ は、読み出した行列 $(A^{p_i})$ を $k_i$ 回繰り返して乗算することによって得られる。

【0052】すなわち、(5)式の演算は、 $M = k_0 + k_1 + \dots + k_n$ 回の乗算によって得られる。得られた演算結果は、レジスタ5に格納され、出力される。

☆  $1+1+6=8$ であるから、8回の乗算で所望の結果が得られることとなる。

【0053】従来は、行列 $A$ の乗算を $m$ 回繰り返していたことを考慮すれば、演算を大幅に減らせることが分かる。例えば、上述した $m=81$ の場合、(4)式より、☆

【0054】本実施例では、  
【0055】  
[外14]

演算回路6に供給すべき行列 $A^{p_i}$ を選択スイッチ8で選ぶように構成したが、

【0056】制御回路9からメモリ回路7にアドレス情報を供給して、この行列を選ぶようにすれば、選択スイッチ8は不要となる。この場合は、制御回路9が選択手段としての機能ももつこととなる。

◆ し、その場合には、予め計算する行列の個数が多くなり、メモリ回路7の容量を増やすことが必要である。なお、メモリ回路7から演算回路6に供給する行列を、  
【0058】  
[外15]

【0057】また、本実施例では、 $t$ の値を増やすことによって、演算をより高速にすることができる。しか ◆

例えば $A^{p_i}$ に固定すれば、 $p_i$ ごとの状態ベクトルを得ることができる。

【0059】また、本実施例では演算回路6を1つとし、各種の行列の中の1つをメモリ回路7から読み出してセットするようにしたが、各種の行列に対応する複数の演算回路をあらかじめ用意し、それらを組み合わせて演算を行うことも可能である。

40 符号の状態を保持するレジスタと、このレジスタに対して演算を施す演算回路と、その演算結果をレジスタに帰還することにより逐次的に符号系列を発生する符号系列発生器において、演算回路の演算手順を変更しながら繰り返し使用することにより、高速にレジスタ状態を設定し、任意の符号系列位相から符号系列を発生させることができる。

【0060】さらに、本実施例によれば、レジスタ状態を任意の状態に高速に設定できる。したがって、短時間で任意の符号系列位相を実現し、その位相から符号系列を発生させることができる。また、選択スイッチ8を特定の行列に固定すれば、通常周波数のクロック信号によって、任意の符号間隔の符号系列を発生することができる。したがって、本発明は、符号系列長が長く、大きく離れたレジスタ状態を設定する場合に特に有効である。

【0062】さらに、本発明によれば、クロック速度を変えずに任意の符号間隔の符号系列を発生させることができる。

【0063】従って、本発明は、符号系列長が長く、大きく離れたレジスタ状態の設定の場合に特に大きな効果がある。

【0061】

【図面の簡単な説明】

【発明の効果】以上説明したように、本発明によれば、

50 【図1】従来の周期15のM系列発生器の具体例を示す

ブロック図である。

【図2】従来の周期15のM系列発生器の一般的構成を示すブロック図である。

【図3】本発明による符号系列発生器の一実施例を示すブロック図である。

【符号の説明】

1 シフトレジスタ回路

2 排他的論理和回路

\* 3 レジスタ

4 演算回路

5 レジスタ

6 演算回路

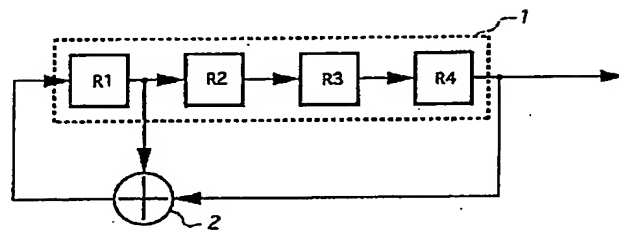
7 メモリ回路

8 選択スイッチ

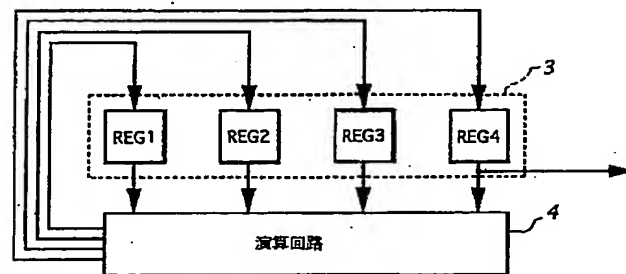
9 制御回路

\*

【図1】



【図2】



【図3】

